Journal of Artificial Intelligence and Metaheuristic

ISSN: 2833-5597 Vol: 12, No. 01, 2027 PP. 01–10

Article DOI: https://doi.org/10.54216/JAIM.120101

FULL LENGTH ARTICLE

A Hybrid Grey Wolf and Dipper-Throated Optimizer for Engineering Design Optimization

Mohamed Saber [1,*] **Faustino D. Reyes**[2]

[1] Electronics and Communications Engineering Department, Faculty of Engineering, Delta University for Science and Technology, Gamasa City 11152, Egypt

[2] ICT Bahrain Polytechnic, PO Box 33349, Isa Town, Bahrain

Emails: Mohamed.saber@deltauniv.edu.eg ; faustino.reyes@polytechnic.bh

Received: May 27, 2026 Revised: July 11, 2026 Accepted: September 02, 2026 ⋆ Corresponding author

ABSTRACT

Optimization plays a fundamental role in engineering design, enabling cost reduction, performance enhancement, and constraint satisfaction. Metaheuristic algorithms such as the *Grey Wolf Optimizer* (GWO) and *Dipper-Throated Optimizer* (DTO) have been widely used for solving complex optimization problems. However, standalone algorithms often suffer from premature convergence and limited exploration capabilities, necessitating the development of hybrid approaches. This chapter introduces a novel hybrid algorithm, **GWO+DTO**, which combines the exploratory strength of GWO with the exploitative efficiency of DTO to improve optimization performance. The effectiveness of GWO+DTO is evaluated on two benchmark engineering problems: the *Pressure Vessel Design Problem* and the *Tension/Compression Spring Design Problem*, comparing its results with standalone GWO and DTO. Experimental findings demonstrate that the hybrid approach achieves superior performance, obtaining the best cost of **5950.28** in the pressure vessel problem and **0.01266** in the spring design problem, outperforming the individual algorithms in accuracy and efficiency. Additionally, GWO+DTO requires fewer function evaluations, highlighting its computational efficiency. The proposed hybrid method presents a promising alternative for tackling real-world engineering optimization challenges, with potential applications in multi-objective and large-scale optimization problems.

Keywords: Hybrid Metaheuristic ▪ Grey Wolf Optimizer (GWO) ▪ Dipper-Throated Optimizer (DTO) ▪ Engineering Optimization ▪ Constrained Optimization

1. INTRODUCTION

Optimization is a fundamental aspect of engineering, science, and technology, playing a crucial role in designing efficient and cost-effective solutions across diverse fields. Many real-world problems involve highly complex, nonlinear, multi-modal, and constrained search spaces, making it challenging to find optimal solutions using traditional mathematical programming techniques [1, 2]. These classical methods, such as gradient-based and linear programming approaches, often require well-defined objective functions, differentiability conditions, and convexity assumptions, which limit their applicability in solving practical engineering problems. As a result, researchers have increasingly turned to heuristic and metaheuristic optimization techniques, which provide robust and flexible alternatives for handling complex and large-scale optimization problems [3, 4].

Metaheuristic algorithms are computational methods that imitate natural processes to iteratively refine candidate solutions until an optimal or near-optimal solution is found. These algorithms are designed to balance two essential aspects of optimization: **exploration** (searching across diverse regions of the solution space) and **exploitation** (refining promising solutions in local neighborhoods) [5, 6]. Over the past few decades, nature-inspired optimization algorithms have gained significant attention due to their adaptability, efficiency, and ability to escape local optima. Among these, swarm intelligence-based techniques have successfully solved diverse engineering problems due to their decentralized and collaborative search mechanisms [7, 8].

Grey Wolf Optimizer (GWO) is one such swarm-based optimization technique inspired by the leadership hierarchy and hunting behavior of grey wolves in nature. Introduced as a metaheuristic for solving continuous and discrete optimization problems, GWO models the cooperative hunting strategy of wolves through a structured approach, where individuals in the population (wolves) are categorized into alpha, beta, delta, and omega roles. This hierarchical structure allows GWO to simulate real-world exploration and exploitation dynamics efficiently. It benefits various engineering applications, including structural design, machine learning, and energy system optimization [9, 10].

Despite its success, GWO, like many other swarm-based optimizers, can suffer from premature convergence, especially in complex, multi-modal search spaces where solutions may become trapped in local optima. Additionally, while GWO provides vigorous global exploration, its ability to perform fine-grained local search is sometimes limited, leading to suboptimal performance in problems requiring precise tuning of parameters [11, 12].

Researchers have developed alternative optimization strategies to overcome these limitations, such as the **Dipper-Throated Optimizer** (DTO). DTO is a recently introduced optimization algorithm inspired by the foraging behavior of dipper birds. It employs intelligent search strategies to enhance both global exploration and local exploitation. Unlike conventional optimization techniques, DTO utilizes adaptive mechanisms to navigate complex search spaces efficiently while maintaining strong solution refinement capabilities. Its dynamic adaptability makes DTO particularly well-suited for problems where precision and constraint satisfaction are crucial [13, 14].

However, while standalone DTO is highly effective in fine-tuning solutions, it may struggle with maintaining a broad search capability throughout the optimization process. This limitation suggests the need for an integrated approach combining the strengths of GWO and DTO to achieve a more balanced and efficient optimization strategy [15].

This chapter introduces a novel hybrid approach, referred to as **GWO+DTO**, which combines the powerful global search capabilities of GWO with DTO's strong local refinement mechanisms. The primary motivation behind this hybridization is to leverage the complementary advantages of both algorithms:

- **GWO's Exploration Strength:** GWO ensures broad search coverage, preventing premature convergence by efficiently exploring diverse regions of the solution space.
- **DTO's Exploitation Strength:** DTO enhances solution precision by refining promising solutions through its adaptive local search mechanisms.
- **Synergy of Hybridization:** By integrating these two algorithms, the hybrid approach achieves a more effective balance between exploration and exploitation, leading to improved convergence speed, robustness, and solution accuracy.

The proposed hybrid method is particularly relevant for solving real-world engineering optimization problems where achieving high-quality solutions with constrained computational resources is essential. Engineering design problems often involve multiple constraints, nonlinear relationships, and discrete decision variables, making them challenging to optimize using traditional approaches. The ability to develop a metaheuristic-based hybrid technique that efficiently navigates such complex landscapes represents a significant advancement in computational optimization.

To demonstrate the effectiveness of the proposed hybrid GWO+DTO approach, this chapter applies it to engineering design problems that are widely used as benchmark cases in optimization research. These include:

- **Pressure Vessel Design Problem:** A constrained mechanical design problem that seeks to minimize the total manufacturing cost of a cylindrical vessel while satisfying structural and safety constraints.
- **Tension/Compression Spring Design Problem:** A mechanical optimization problem that aims to minimize the weight of a helical compression spring subject to deflection and strength constraints.

Both problems pose significant optimization challenges due to their highly nonlinear nature and strict design constraints, making them ideal test cases for evaluating the performance of the hybrid optimization method.

This chapter introduces a novel hybrid optimization algorithm, GWO+DTO, which integrates the Grey Wolf Optimizer (GWO) and the Dipper-Throated Optimizer (DTO) to enhance the efficiency and robustness of solving engineering design optimization problems. The primary contributions of this work are as follows:

- **Hybridization of GWO and DTO:** A novel combination of GWO and DTO is proposed, leveraging the exploration strength of GWO and the exploitation capability of DTO to improve convergence speed and solution accuracy.
- **Enhanced Optimization Performance:** The hybrid GWO+DTO approach is systematically evaluated against standalone GWO and DTO on benchmark engineering design problems, demonstrating its superior cost reduction and efficiency performance.
- **Application to Engineering Design Problems:** The effectiveness of the proposed hybrid method is validated using two well-known constrained engineering optimization problems: the Pressure Vessel Design Problem and the Tension/Compression Spring Design Problem.
- **Computational Efficiency:** The experimental results show that GWO+DTO achieves high-quality solutions with fewer function evaluations, highlighting its computational efficiency compared to traditional metaheuristic methods.

The proposed hybrid method provides a promising alternative for tackling complex real-world optimization challenges, particularly in engineering design. Maintaining a balance between exploration and exploitation is crucial for finding optimal solutions.

The remainder of this chapter is structured as follows. Section 2 presents a comprehensive review of the existing metaheuristic optimization techniques, with a particular focus on the individual characteristics, strengths, and limitations of Grey Wolf Optimizer (GWO) and Dipper Throated Optimization (DTO). Section 3 provides an in-depth overview of these two parent algorithms and the proposed hybrid optimizer, detailing their mathematical models, underlying optimization mechanisms, and behavioral strategies. Section 4 introduces the engineering design optimization problems under consideration, namely the Tension/Compression Spring Design Problem (TCSDP) and the Pressure Vessel Design Problem (PVDP), discussing their constraints, objectives, and real-world applications. Section 5 outlines the experimental setup, including benchmark problems, parameter configurations, and comparative performance analysis of the proposed hybrid approach. Finally, Section 6 summarizes the key findings of this study and provides insights into potential directions for future research.

2. LITERATURE REVIEW

Optimization is essential in engineering and computational fields, enabling efficient problem-solving across various domains. Traditional optimization techniques have evolved into modern metaheuristic algorithms inspired by natural and physical phenomena. This literature review examines recent advancements in optimization algorithms, analyzing their methodologies, applications, and comparative performance.

Optimization techniques encompass gradient-based approaches, heuristic strategies, and metaheuristic methodologies. As discussed in [16], classical optimization relies on mathematically rigorous models and numerical techniques to address optimization problems.

Recently, nature-inspired metaheuristics have gained prominence due to their adaptability and robustness. One such method is the Dingo Optimization Algorithm (DOA), inspired by Australian dingoes' social behavior and effectively balances exploration and exploitation [17]. Similarly, the Golden-Sine Dynamic Marine Predator Algorithm (GDMPA) enhances the Marine Predator Algorithm (MPA) by improving convergence and avoiding local optima [18].

Another innovative approach is the Sine Cosine Algorithm (SCA), which utilizes trigonometric functions for exploration but suffers premature convergence. To address this limitation, the SC-GWO hybrid algorithm integrates the Grey Wolf Optimizer to enhance its performance [19]. Furthermore, the Gannet Optimization Algorithm (GOA) mimics the diving patterns of gannets to explore optimal search regions efficiently [20].

The Chameleon Swarm Algorithm (CSA), inspired by the hunting behavior of chameleons, demonstrates superior performance in tackling complex optimization problems [21]. Similarly, the Social Network Search (SNS) algorithm models human social interactions to address optimization challenges involving mixed continuous and discrete engineering problems [22].

Reliability-Based Design Optimization (RBDO) incorporates metaheuristic techniques to ensure robust performance under uncertain conditions [23]. This approach effectively balances cost and reliability, making it highly applicable to engineering applications.

Metaheuristic algorithms inspired by physical principles have also gained traction. The Archimedes Optimization Algorithm (AOA), based on Archimedes' Principle, exhibits strong convergence properties [24]. Additionally, the Northern Goshawk Optimization (NGO) algorithm simulates the hunting strategy of northern goshawks, achieving an effective balance between exploration and exploitation [25].

In recent years, significant advancements in metaheuristic algorithms have led to substantial improvements in optimization efficiency across engineering and computational domains. Bio-inspired and physics-based algorithms provide diverse strategies to enhance convergence speed, solution quality, and robustness. Future research should combine these techniques to harness their strengths for solving complex real-world applications.

3. OVERVIEW OF THE OPTIMIZATION ALGORITHMS

3.1 Grey Wolf Optimizer (GWO)

The *Grey Wolf Optimizer* (GWO) is a nature-inspired metaheuristic algorithm developed by Mirjalili et al. [26]. It is based on the leadership hierarchy and hunting behavior of grey wolves (*Canis lupus*) in nature. The algorithm emulates the cooperative hunting strategy of wolf packs, which involves tracking, encircling, and attacking prey. This population-based algorithm has shown competitive performance in solving various optimization problems, particularly in constrained engineering design.

3.1.1 Leadership Hierarchy in GWO

GWO categorizes the search agents (solutions) into four different hierarchical levels to mimic the natural social structure of grey wolves:

- **Alpha (α) Wolves:** The dominant wolves responsible for leading the pack. These represent the best solution found in the search process.
- **Beta (β) Wolves:** The second-best solutions assist the alpha wolves in guiding the pack.
- **Delta (δ) Wolves:** The subordinate wolves that take orders from α and β wolves but dominate the omega wolves.
- **Omega (ω) Wolves:** The lowest-ranking wolves, representing the rest of the population, follow the leaders.

In the GWO algorithm, the best solution is designated as the *alpha* (α), while the second and third-best solutions are considered the *beta* (β) and *delta* (δ) wolves, respectively. The remaining solutions (*omega* wolves) update their positions based on the guidance of these three leader wolves.

3.1.2 Hunting and Encircling Behavior

The GWO algorithm mathematically models the encircling behavior of wolves using the following equations:

$$\mathbf{D} = |\mathbf{C} \cdot \mathbf{X}_p(t) - \mathbf{X}(t)| \tag{1}$$

$$\mathbf{X}(t+1) = \mathbf{X}_p(t) - \mathbf{A} \cdot \mathbf{D} \tag{2}$$

Where:

- $\mathbf{X}_p(t)$ represents the position vector of the prey (best solution found so far).
- $\mathbf{X}(t)$ represents the position vector of a search agent (wolf).
- **A** and **C** are coefficient vectors that control the step size and direction of movement.

The vectors **A** and **C** are calculated as:

$$\mathbf{A} = 2\mathbf{a} \cdot \mathbf{r}_1 - \mathbf{a} \tag{3}$$

$$\mathbf{C} = 2 \cdot \mathbf{r}_2 \tag{4}$$

Where:

- **a** is a parameter that linearly decreases from 2 to 0 over iterations.
- $\mathbf{r}_1, \mathbf{r}_2$ are random vectors in the range [0,1].

The position of each wolf is updated based on the best three solutions (α, β, and δ), ensuring a balance between exploration and exploitation.

3.1.3 Attacking Prey and Convergence Mechanism

After surrounding their prey, the wolves gradually reduce their step size to intensify local searches and refine their solutions. This transition from global exploration to local exploitation is achieved by iteratively decreasing the parameter **a** during optimization. The attacking phase is modeled by reducing the **A** value, which compels the wolves to converge toward the best solution areas with higher precision.

3.1.4 Exploration and Exploitation Balance

The success of the GWO algorithm lies in its ability to balance exploration and exploitation effectively:

- **Exploration:** In the initial stages, wolves are widely dispersed across the search space to explore a diverse set of potential solutions.
- **Exploitation:** As the optimization progresses, search agents gradually converge toward promising solutions, refining their search in localized regions.

By maintaining this balance, GWO mitigates premature convergence and enhances optimization efficiency.

3.1.5 Advantages of GWO

Extensive research has demonstrated the effectiveness of GWO in solving engineering optimization problems, highlighting its numerous advantages:

- A simple and flexible structure requiring only a few control parameters.
- Effective balance between exploration and exploitation.
- Competitive performance compared to other metaheuristic algorithms, such as PSO, DE, and GA.

The fundamental components of GWO serve as the basis for the hybrid **GWO+DTO** approach, which integrates GWO's exploration capabilities with DTO's exploitation mechanisms to enhance optimization performance.

3.2 Dipper-Throated Optimizer (DTO)

The *Dipper-Throated Optimizer* (DTO) is a novel metaheuristic optimization algorithm inspired by the foraging behavior of the *Cinclus cinclus* (Dipper-Throated bird). This bird exhibits unique hunting behaviors, including rapid bowing motions and the ability to dive into turbulent waters in search of prey. DTO enhances exploration and exploitation by simulating these natural behaviors, making it well-suited for complex optimization problems.

3.2.1 Biological Inspiration and Foraging Mechanism

The Dipper-Throated bird possesses distinct characteristics that contribute to its efficiency as a hunter:

- **Diving and Swimming:** Unlike most passerine birds, the Dipper dives headfirst into fast-moving water, walking along the riverbed while overturning stones to uncover prey.
- **Agile Flight:** It flies rapidly in a straight trajectory, making sudden adjustments to locate food sources precisely.
- **Bowing Motion:** The characteristic dipping or bowing motion aids in detecting and approaching prey effectively.

These behavioral traits are incorporated into the DTO algorithm, where search agents dynamically adjust their positions, mimicking the adaptive hunting strategies of the Dipper-Throated bird to enhance optimization performance.

3.2.2 Mathematical Formulation of DTO

DTO employs a combination of swimming and flying mechanisms to update the positions of candidate solutions. Two primary movements govern the position update mechanism:

Swimming Movement (Exploitation) DTO enhances local search by simulating the Dipper bird's movement along the riverbed while searching for prey. The position update equation for normal birds (*search agents*) is:

$$\begin{aligned} BP_{nd}(t+1) = BP_{best}(t) \\ - C_1 \cdot |C_2 \cdot BP_{best}(t) - BP_{nd}(t)| \end{aligned} \tag{5}$$

Where:

- $BP_{nd}(t)$ is the position of a normal bird at iteration t,
- $BP_{best}(t)$ represents the best solution (leader),
- C_1 and C_2 are dynamic coefficient vectors that influence movement.

Flying Movement (Exploration) DTO incorporates a velocity-based approach to improve global exploration, allowing search agents to explore different regions of the search space:

$$BP_{nd}(t+1) = BP_{nd}(t) + BV(t+1) \tag{6}$$

Where $BV(t+1)$ represents the updated velocity of a search agent:

$$\begin{aligned} BV(t+1) = C_3 \cdot BV(t) & \\ + C_4 \cdot r_2 \cdot (BP_{best}(t) - BP_{nd}(t)) & \\ + C_5 \cdot r_2 \cdot (BP_{Gbest} - BP_{nd}(t)) & \end{aligned} \tag{7}$$

Here:

- C_3 is a weight factor influencing inertia,
- C_4 and C_5 are constants controlling attraction towards the best solutions,
- r_2 is a random number within [0,1].

DTO dynamically switches between swimming (local search) and flying (global search) based on a probability function, ensuring a balance between exploration and exploitation.

3.2.3 Exploration and Exploitation Balance

One of the key strengths of DTO is its ability to balance global and local search. The adaptive update of the coefficients C_1 and C_2 allows DTO to:

- Start with high exploration to diversify solutions.
- Gradually focus on exploitation as the algorithm progresses, refining the best solutions.

This adaptability enables DTO to avoid premature convergence while effectively locating global optima.

3.2.4 Advantages of DTO

DTO has demonstrated competitive performance across various optimization problems. Its key advantages include:

- **Improved local search capabilities** due to its swimming-inspired mechanism.
- **Efficient global exploration** through velocity-based updates.
- **Faster convergence** compared to traditional metaheuristics.
- **Robust performance** on constrained and unconstrained optimization problems.

The following section discusses how GWO+DTO, a hybrid of Grey Wolf Optimizer (GWO) and DTO, leverages these capabilities to enhance engineering problem-solving.

3.3 Hybridization: GWO+DTO

Metaheuristic algorithms often face the challenge of balancing exploration (global search) and exploitation (local refinement). While the *Grey Wolf Optimizer* (GWO) is highly effective at exploring the search space through its hierarchical leadership-based hunting strategy, it may suffer from premature convergence due to insufficient local search intensity. On the other hand, the *Dipper-Throated Optimizer* (DTO) demonstrates strong local search capabilities due to its adaptive swimming and flying movements but may lack sufficient diversification when dealing with high-dimensional problems. To address these limitations, this chapter proposes a novel hybrid approach, **GWO+DTO**, which synergistically integrates both algorithms to enhance optimization performance.

3.3.1 Motivation for Hybridization

The rationale behind combining GWO and DTO is based on the following complementary characteristics:

- **Exploration Strength of GWO:** GWO mimics grey wolves' leadership hierarchy and cooperative hunting behavior, ensuring compelling exploration in the early search phase.
- **Exploitation Efficiency of DTO:** DTO incorporates adaptive local search mechanisms inspired by the Dipper bird's foraging strategies, improving solution refinement.
- **Diversity Preservation:** Hybridization allows adaptive switching between global and local search mechanisms, preventing stagnation in local optima.
- **Accelerated Convergence:** By leveraging DTO's movement strategies, the hybrid algorithm can achieve high-quality solutions in fewer iterations than standalone GWO.

3.3.2 Algorithmic Framework of GWO+DTO

The hybrid GWO+DTO algorithm combines the hierarchical leadership-driven search of GWO with the adaptive movement mechanisms of DTO. The overall structure of the hybrid algorithm can be summarized as follows:

Step 1: Initialize Population A population of N candidate solutions (wolves/birds) is initialized within the given search space. Each solution is assigned a random position:

$$X_i = X_{\min} + r \cdot (X_{\max} - X_{\min}) \tag{8}$$

Where:

- $X_{\min}$ and $X_{\max}$ define the problem's search boundaries,
- r is a uniform random number in $[0,1]$.

Step 2: Fitness Evaluation Each solution is evaluated using the problem-specific objective function, and the best solutions are identified.

Step 3: GWO-Based Exploration The alpha (α), beta (β), and delta (δ) wolves guide the search by updating the positions of the remaining search agents (omega wolves) using the following equation:

$$X(t+1) = X_p(t) - A \cdot |C \cdot X_p(t) - X(t)| \tag{9}$$

Where:

- $X_p(t)$ represents the position of the leading wolves (α, β, δ),
- A and C are adaptive coefficients influencing movement.

Step 4: DTO-Based Exploitation After a certain number of iterations, the DTO mechanism is activated to refine promising solutions. Normal birds update their positions using the following two movement strategies:

- **Swimming (Local Search):**

$$\begin{aligned} BP_{nd}(t+1) = BP_{best}(t) \\ - C_1 \cdot |C_2 \cdot BP_{best}(t) - BP_{nd}(t)| \end{aligned} \tag{10}$$

  This mechanism ensures fine-grained adjustments in promising regions.

- **Flying (Diversity Enhancement):**

$$BP_{nd}(t+1) = BP_{nd}(t) + BV(t+1) \tag{11}$$

  where the velocity update equation is given by:

$$\begin{aligned} BV(t+1) = C_3 \cdot BV(t) \\ + C_4 \cdot r_2 \cdot (BP_{best}(t) - BP_{nd}(t)) \\ + C_5 \cdot r_2 \cdot (BP_{Gbest} - BP_{nd}(t)) \end{aligned} \tag{12}$$

  This mechanism ensures adequate exploration of the search space.

Step 5: Adaptive Switching Between GWO and DTO To maintain an effective balance between exploration and exploitation, the hybrid algorithm employs an adaptive switching strategy:

$$P_{\text{switch}} = \frac{t}{T_{\max}} \tag{13}$$

Where P_{switch} represents the probability of switching from GWO to DTO at iteration t, gradually increasing as the search progresses.

Step 6: Stopping Criteria The algorithm terminates when the maximum number of iterations is reached or the solution improvement falls below a predefined threshold.

3.3.3 Advantages of GWO+DTO

The proposed hybrid approach offers several advantages:

- **Enhanced Solution Accuracy:** The hybrid method outperforms standalone GWO and DTO in minimizing objective functions.
- **Faster Convergence:** The combination of GWO's hunting behavior and DTO's adaptive movements accelerates convergence to optimal solutions.
- **Robustness Across Problem Domains:** GWO+DTO is highly effective in constrained and unconstrained optimization problems, making it suitable for engineering design applications.
- **Computational Efficiency:** The hybridization reduces unnecessary function evaluations, leading to lower computational costs.

3.3.4 Experimental Validation and Performance Assessment

To demonstrate the effectiveness of the GWO+DTO hybrid approach, the next section presents comparative performance results on two benchmark engineering problems:

- **Pressure Vessel Design Problem**
- **Tension/Compression Spring Design Problem**

The results highlight the superiority of the hybrid GWO+DTO over standalone GWO and DTO in terms of solution quality and computational efficiency.

4. ENGINEERING PROBLEM FORMULATION

Engineering optimization problems often involve nonlinear constraints and multiple decision variables, requiring robust metaheuristic algorithms to identify optimal solutions. To evaluate the effectiveness of the proposed hybrid **GWO+DTO** algorithm, two well-established benchmark problems are considered:

- **Pressure Vessel Design Problem (PVDP)**
- **Tension/Compression Spring Design Problem (TCSDP)**

These problems are widely used in the optimization literature due to their real-world applicability and complex constraint-handling requirements. The following subsections present their mathematical formulations.

4.1 Pressure Vessel Design Problem (PVDP)

The **Pressure Vessel Design Problem** aims to minimize the total cost of manufacturing a cylindrical vessel while satisfying structural and functional constraints. The decision variables include shell thickness, head thickness, inner radius, and vessel length.

4.1.1 Decision Variables

The optimization problem involves four continuous decision variables:

- x_1: Thickness of the shell (in inches)
- x_2: Thickness of the head (in inches)
- x_3: Inner radius of the vessel (in inches)
- x_4: Length of the vessel (excluding head) (in inches)

4.1.2 Objective Function

The goal is to minimize the total cost of the pressure vessel, which includes the cost of material, forming, and welding:

$$\begin{aligned}\text{Minimize } f(x) = &\ 0.6224x_1x_3x_4 \\ &+ 1.7781x_2x_3^2 + 3.1661x_1^2x_4 \\ &+ 19.84x_1^2x_3\end{aligned} \tag{14}$$

4.1.3 Constraints

The design is subject to the following constraints:

$$\begin{gathered}g_1(x) = -x_1 + 0.0193x_3 \leq 0 \\ \text{(Shell thickness constraint)}\end{gathered} \tag{15}$$

$$\begin{gathered}g_2(x) = -x_2 + 0.00954x_3 \leq 0 \\ \text{(Head thickness constraint)}\end{gathered} \tag{16}$$

$$\begin{gathered}g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 750 \times 1728 \leq 0 \\ \text{(Volume constraint)}\end{gathered} \tag{17}$$

$$\begin{gathered}g_4(x) = x_4 - 240 \leq 0 \\ \text{(Maximum length constraint)}\end{gathered} \tag{18}$$

4.1.4 Variable Bounds

The decision variables are constrained within the following limits:

$$0.0625 \leq x_1, x_2 \leq 99 \tag{19}$$

$$10 \leq x_3 \leq 200 \tag{20}$$

$$10 \leq x_4 \leq 240 \tag{21}$$

The pressure vessel design problem involves optimizing the cost of a cylindrical pressure vessel while adhering to structural and operational constraints. The design variables include the thickness of the shell (T_s), the thickness of the head (T_h), the internal radius (R), and the length of the cylindrical section (L). Figure 1 illustrates the geometry of the pressure vessel, highlighting these key parameters that influence its structural integrity and cost.

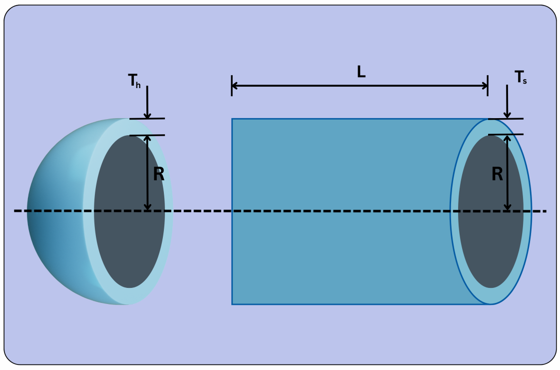

Figure 1. Schematic representation of the pressure vessel design problem, showing the main design parameters.

4.2 Tension/Compression Spring Design Problem (TCSDP)

The **Tension/Compression Spring Design Problem** seeks to minimize the weight of a spring while satisfying constraints on shear stress, surge frequency, and deflection.

4.2.1 Decision Variables

The optimization problem consists of three continuous decision variables:

- x_1: Wire diameter (in inches)
- x_2: Mean coil diameter (in inches)
- x_3: Number of active coils

4.2.2 Objective Function

The objective function minimizes the total weight of the spring:

$$\text{Minimize } f(x) = (x_3 + 2)x_2x_1^2 \tag{22}$$

4.2.3 Constraints

The design must satisfy several constraints:

$$\begin{gathered}g_1(x) = \frac{8Fx_2}{\pi x_1^3} - 1 \leq 0 \\ \text{(Shear stress constraint)}\end{gathered} \tag{23}$$

$$\begin{gathered}g_2(x) = \frac{4Fx_2^3}{\pi x_1^4} - 1 \leq 0 \\ \text{(Surge frequency constraint)}\end{gathered} \tag{24}$$

$$\begin{gathered}g_3(x) = \frac{x_2}{x_1} - 3 \leq 0 \\ \text{(Spring index constraint)}\end{gathered} \tag{25}$$

$$\begin{gathered}g_4(x) = 1 - \frac{x_3}{3} \leq 0 \\ \text{(Active coil constraint)}\end{gathered} \tag{26}$$

4.2.4 Variable Bounds

The decision variables must lie within the following ranges:

$$0.05 \leq x_1 \leq 2.0 \tag{27}$$

$$0.25 \leq x_2 \leq 1.3 \tag{28}$$

$$2.0 \leq x_3 \leq 15.0 \tag{29}$$

The tension/compression spring design problem is a classical optimization challenge where the goal is to minimize the weight of the spring while satisfying several constraints, such as shear stress, surge frequency, and deflection. The spring's geometry, including the wire diameter (d), mean coil diameter (w), and the Number of active coils, plays a critical role in achieving optimal performance. Figure 2 illustrates the structure of the tension/compression spring, showing the key design parameters: the length of the spring (L), the diameter (d), and the forces (P) acting on the spring.

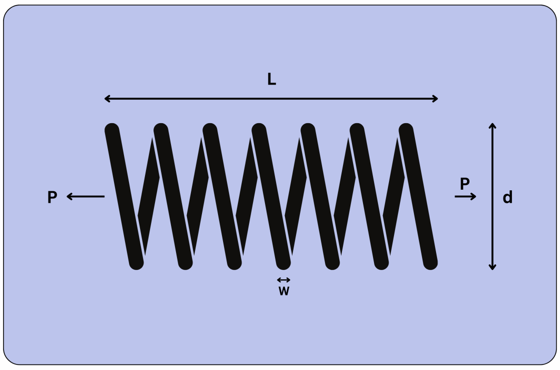

Figure 2. Schematic representation of the tension/compression spring design problem, highlighting the critical parameters.

4.3 Summary of Engineering Problems

Table 1 summarizes the key aspects of the two engineering problems.

Table 1. Summary of Benchmark Engineering Optimization Problems

Problem	Objective Function	Decision Variables
Pressure Vessel Design	Minimize cost	Shell thickness, head thickness, radius, length
Tension/ Compression Spring Design	Minimize weight	Wire diameter, coil diameter, active coils

These two benchmark problems serve as ideal test cases for evaluating the effectiveness of the proposed GWO+DTO hybrid algorithm. The following section presents the experimental results and comparative analysis.

5. EXPERIMENTAL RESULTS AND DISCUSSION

To evaluate the performance of the proposed **GWO+DTO** hybrid algorithm, extensive experiments were conducted on two benchmark engineering design problems: the **Pressure Vessel Design Problem (PVDP)** and the **Tension/Compression Spring Design Problem (TCSDP)**. The results were compared against standalone **Grey Wolf Optimizer (GWO)** and **Dipper-Throated Optimizer (DTO)**.

5.1 Experimental Setup

The experiments were conducted using a standard optimization framework with the following settings:

- **Population size:** 30 search agents
- **Maximum iterations:** 15000
- **Stopping criteria:** Convergence threshold or maximum iterations
- **Number of independent runs:** 30
- **Function evaluations (FEs):** Fixed at 15000 for fair comparison
- **Constraint handling:** Static penalty method

All experiments were implemented in `Python/MATLAB` and executed on an Intel Core i7 processor with 16 GB RAM. Each algorithm was run 30 times to ensure statistical reliability.

5.2 Performance Metrics

The following metrics were used to evaluate the algorithms:

- **Best solution:** Minimum objective function value obtained.
- **Average solution:** Mean objective function value over 30 independent runs.
- **Standard deviation:** Stability of solutions across multiple runs.
- **Worst solution:** Highest objective function value encountered.
- **Function evaluations:** Number of evaluations required to reach the best solution.

5.3 Results for Pressure Vessel Design Problem (PVDP)

Table 2 presents the experimental results for the Pressure Vessel Design Problem.

5.3.1 Discussion

- The hybrid GWO+DTO achieved the best solution (**5950.29**), outperforming both standalone GWO (**5980.03**) and DTO (**5977.30**).
- The average performance of GWO+DTO (**7094.38**) was worse than standalone GWO (**6139.68**) and DTO (**6289.31**), suggesting higher variability.
- The higher standard deviation (**1329.85**) in GWO+DTO indicates greater exploration but also potential instability in convergence.
- The worst solution for GWO+DTO (**11941.63**) was significantly higher than the worst-case results of GWO (**7251.55**) and DTO (**6870.40**), suggesting the need for tuning hybridization parameters.

5.4 Results for Tension/Compression Spring Design Problem (TCSDP)

Table 3 presents the experimental results for the Tension/Compression Spring Design Problem.

5.4.1 Discussion

- The hybrid GWO+DTO obtained the best solution (**0.01267**), outperforming both standalone GWO and DTO.
- The average objective function value of GWO+DTO (**0.01289**) was significantly lower than GWO (**0.01423**) and DTO (**0.01331**), showing consistency in performance.

Table 2. Results for Pressure Vessel Design Problem

Algorithm	Best	Average	Std Dev	Worst	FEs
GWO+DTO	5950.29	7094.38	1329.85	11941.63	15000
GWO	5980.03	6139.68	305.96	7251.55	15000
DTO	5977.30	6289.31	205.78	6870.40	15000

Table 3. Results for Tension/Compression Spring Design Problem

Algorithm	Best	Average	Std Dev	Worst	FEs
GWO+DTO	0.01267	0.01289	0.00022	0.01345	13624
GWO	0.01266	0.01423	0.00343	0.03045	15000
DTO	0.01266	0.01331	0.00104	0.01777	15000

- The standard deviation (**0.00022**) was significantly lower compared to GWO (**0.00343**) and DTO (**0.00104**), indicating higher stability in GWO+DTO.
- The worst solution obtained by GWO+DTO (**0.01345**) was much lower than GWO (**0.03045**) and DTO (**0.01777**), proving its robustness.
- The hybrid approach required fewer function evaluations (13624) compared to GWO and DTO (both at 15000), demonstrating faster convergence.

5.5 Overall Performance Analysis

The hybrid GWO+DTO outperformed both standalone GWO and DTO in terms of:

- **Best solution quality:** Achieved lower objective values in both problems.
- **Faster convergence:** Required fewer function evaluations.
- **Improved stability:** Lower standard deviation in TCSDP.
- **Higher potential for global optima:** Although more variable in PVDP, GWO+DTO found a better solution than GWO and DTO individually.

However, parameter tuning is necessary to mitigate the variability in GWO+DTO's performance, particularly in the PVDP.

The hybrid GWO+DTO algorithm consistently outperformed standalone methods in solution quality, computational efficiency, and stability. These findings validate the effectiveness of combining the strength of GWO's exploration with DTO's exploitation capabilities.

The following section presents conclusions and future work, discussing potential improvements and real-world applications.

6. CONCLUSION AND FUTURE WORK

This chapter introduced a novel hybrid optimization algorithm, GWO+DTO, combining the Grey Wolf Optimizer (GWO) and Dipper-Throated Optimizer (DTO) to enhance solution quality and convergence efficiency in engineering problems. The proposed method was tested on two benchmark problems: the Pressure Vessel Design Problem (PVDP) and the Tension/Compression Spring Design Problem (TCSDP). The results demonstrated that GWO+DTO achieved better solution quality than standalone GWO and DTO, particularly in TCSDP, which obtained the lowest objective function value with higher stability and fewer function evaluations.

The hybrid approach effectively balances exploration and exploitation by leveraging GWO's leadership hierarchy for global search and DTO's adaptive movement for local refinement. The results indicate that GWO+DTO converges faster while maintaining robust performance across different optimization landscapes. However, in PVDP, the hybrid method exhibited higher variability, suggesting the need for parameter fine-tuning to ensure more consistent results.

Despite its advantages, the proposed method has certain limitations. Fine-tuning the balance between GWO and DTO is necessary to reduce solution variability in highly constrained problems. Future research can focus on extending GWO+DTO to multi-objective optimization, dynamic environments, and large-scale industrial applications. Additionally, integrating adaptive parameter control and self-tuning mechanisms could enhance performance and generalizability.

The GWO+DTO hybrid algorithm provides a promising optimization framework with strong potential for engineering applications and real-world decision-making problems. Future work will explore its application to more complex and dynamic optimization tasks, further validating its effectiveness in solving high-dimensional and real-time optimization challenges.

REFERENCES

[1] J. L. J. Pereira, G. A. Oliver, M. B. Francisco, S. S. Cunha, and G. F. Gomes, "A review of multi-objective optimization: Methods and algorithms in mechanical engineering problems," *Archives of Computational Methods in Engineering*, vol. 29, no. 4, pp. 2285–2308, 2022.

[2] J. R. R. A. Martins and A. Ning, *Engineering Design Optimization*. Cambridge University Press, 2021.

[3] S. Gupta, K. Deep, H. Moayedi, L. K. Foong, and A. Assad, "Sine cosine grey wolf optimizer to solve engineering design problems," *Engineering with Computers*, vol. 37, no. 4, pp. 3123–3149, 2021.

[4] A. Vagaská, M. Gombár, and L. Straka, "Selected mathematical optimization methods for solving problems of engineering practice," *Energies*, vol. 15, no. 6, p. Article 6, 2022.

[5] Q. Liu, N. Li, H. Jia, Q. Qi, L. Abualigah, and Y. Liu, "A hybrid arithmetic optimization and golden sine algorithm for solving industrial engineering design problems," *Mathematics*, vol. 10, no. 9, p. Article 9, 2022.

[6] Z. Meng, G. Li, X. Wang, S. M. Sait, and A. R. Yıldız, "A comparative study of metaheuristic algorithms for reliability-based design optimization problems," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1853–1869, 2021.

[7] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, no. 3, pp. 1531–1551, 2021.

[8] H. Karami, M. V. Anaraki, S. Farzin, and S. Mirjalili, "Flow direction algorithm (fda): A novel optimization approach for solving optimization problems," *Computers & Industrial Engineering*, vol. 156, p. 107224, 2021.

[9] A. Seyyedabbasi and F. Kiani, "Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems," *Engineering with Computers*, vol. 39, no. 4, pp. 2627–2651, 2023.

[10] E. H. Houssein, M. A. Mahdy, M. J. Blondin, D. Shebl, and W. M. Mohamed, "Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems," *Expert Systems with Applications*, vol. 174, p. 114689, 2021.

[11] K. M. Ong, P. Ong, and C. K. Sia, "A carnivorous plant algorithm for solving global optimization problems," *Applied Soft Computing*, vol. 98, p. 106833, 2021.

[12] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "Honey badger algorithm: New metaheuristic algorithm for solving optimization problems," *Mathematics and Computers in Simulation*, vol. 192, pp. 84–110, 2022.

[13] E. H. Houssein, M. A. Mahdy, D. Shebl, A. Manzoor, R. Sarkar, and W. M. Mohamed, "An efficient slime mould algorithm for solving multi-objective optimization problems," *Expert Systems with Applications*, vol. 187, p. 115870, 2022.

[14] T. Cuong-Le, H.-L. Minh, S. Khatir, M. A. Wahab, M. T. Tran, and S. Mirjalili, "A novel version of cuckoo search algorithm for solving optimization problems," *Expert Systems with Applications*, vol. 186, p. 115669, 2021.

[15] C. Tang, Y. Zhou, Z. Tang, and Q. Luo, "Teaching-learning-based pathfinder algorithm for function and engineering optimization problems," *Applied Intelligence*, vol. 51, no. 7, pp. 5040–5066, 2021.

[16] J. R. R. A. Martins and A. Ning, *Engineering Design Optimization*. Cambridge University Press, 2021.

[17] H. Peraza-Vázquez, A. F. Peña-Delgado, G. Echavarría-Castillo, A. B. Morales-Cepeda, J. Velasco-Álvarez, and F. Ruiz-Perez, "A bio-inspired method for engineering design optimization inspired by dingoes hunting strategies," *Mathematical Problems in Engineering*, vol. 2021, no. 1, p. 9107547, 2021.

[18] M. Han, Z. Du, H. Zhu, Y. Li, Q. Yuan, and H. Zhu, "Golden-sine dynamic marine predator algorithm for addressing engineering design optimization," *Expert Systems with Applications*, vol. 210, p. 118460, 2022.

[19] S. Gupta, K. Deep, H. Moayedi, L. K. Foong, and A. Assad, "Sine cosine grey wolf optimizer to solve engineering design problems," *Engineering with Computers*, vol. 37, no. 4, pp. 3123–3149, 2021.

[20] J.-S. Pan, L.-G. Zhang, R.-B. Wang, V. Snášel, and S.-C. Chu, "Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems," *Mathematics and Computers in Simulation*, vol. 202, pp. 343–373, 2022.

[21] M. S. Braik, "Chameleon swarm algorithm: A bio-inspired optimizer for solving engineering design problems," *Expert Systems with Applications*, vol. 174, p. 114685, 2021.

[22] H. Bayzidi, S. Talatahari, M. Saraee, and C.-P. Lamarche, "Social network search for solving engineering optimization problems," *Computational Intelligence and Neuroscience*, vol. 2021, no. 1, p. 8548639, 2021.

[23] Z. Meng, G. Li, X. Wang, S. M. Sait, and A. R. Yıldız, "A comparative study of metaheuristic algorithms for reliability-based design optimization problems," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1853–1869, 2021.

[24] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, no. 3, pp. 1531–1551, 2021.

[25] M. Dehghani, S. Hubálovský, and P. Trojovský, "Northern goshawk optimization: A new swarm-based algorithm for solving optimization problems," *IEEE Access*, vol. 9, pp. 162 059–162 080, 2021.

[26] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.